Automatic Summarization of Mouse Gene Information by Clustering and Sentence Extraction from MEDLINE Abstracts

Jianji Yang, M.S., Aaron M. Cohen, M.D., M.S., William Hersh, M.D.
Oregon Health & Science University, Portland, Oregon, USA

Abstract

Tools to automatically summarize gene information from the literature have the potential to help genomics researchers better interpret gene expression data and investigate biological pathways. The task of finding information on sets of genes is common for genomic researchers, and PubMed is still the first choice because the most recent and original information can only be found in the unstructured, free text biomedical literature. However, finding information on a set of genes by manually searching and scanning the literature is a time-consuming and daunting task for scientists. We built and evaluated a query-based automatic summarizer of information on mouse genes studied in microarray experiments. The system clusters a set of genes by MeSH, GO and free text features and presents summaries for each gene by ranked sentences extracted from MEDLINE abstracts. Evaluation showed that the system seems to provide meaningful clusters and informative sentences are ranked higher by the algorithm.

Introduction

With the increasing volume of published free-text scientific articles, even the most robust information retrieval (IR) engine returns more documents and abstracts than biomedical scientists are able to manually review. The problem is aggravated by the information-intensive nature of some "high-throughput" technologies, such as gene microarray experiments that can study expression at a genome-wide scale. Some of the possible approaches to this problem include: document clustering, information extraction, question answering, and summarization.

Document clustering techniques attempt to group a text collection into clusters of articles that relate to a similar topic. PubClust [1] is a system that groups the result of any PubMed search using words in the returned abstracts as features. Therefore, users can pick the topics of interest for their purpose.

Information extraction (IE) methods discover structured information from free text using natural language processing (NLP) techniques. IE is used mostly in the biomedical domain to extract relations between biological entities [2]. IE often involves hand-crafted templates and rules based on expert knowledge and intensive NLP processing.

Question answering has been getting more attention recently. The idea is to let users ask a structured question, such as '*What is the role of prion in mad cow disease?*' and the system will process the document collection and extract the corresponding information as answer. This is similar to IE but is real-time and gives the user more control over the information extracted as well as more context with which to verify the answers. The TREC Genomics Track has recently focused on this topic [3].

Another potentially useful, but less-studied approach is to automatically produce customized summaries for the users who are analyzing the result of a specific microarray experiment. Summarization is defined by Sparck Jones [4] as "*a reductive transformation of source text to summary text through content reduction selection and/or generalization on what is important in the source*". Automatic summarization systems have been studied since the late 1950s [5,6] and applied in different domains such as news, with some notable success [7]. However, adopting the technology in the biomedical domain is not as straightforward. There are fewer resources available in biomedicine, such as test corpora and knowledge bases, which makes training and evaluation more difficult. Summaries for biomedical literature probably require a different focus than news articles [8]. The information that most interests scientists may reside in sentences describing some specific biological processes (use of domain specific language e.g. phosphorylation, activation, co-expression) while in the news domain, the *who, when, what, and where* elements are generally applicable and often the most important. We can exploit the specific requirements in the biomedical domain by emphasizing domain specific keywords to construct summaries.

Here we report an approach to produce gene information summaries by sentence extraction following the Edmundson paradigm [5]. Focusing on gene sets from microarray experiments, the system consists of a two-step process. First the gene set was clustered into functional related groups based on free

text, MeSH, and GO features. Next, a summary for each gene is generated as sentences ranked by domain specific vocabulary, length, representation of its functional cluster, cue words and recency. Finally, each sentence is linked to MEDLINE to allow interested users further investigation. Previous work either focus on functional gene clustering [9,10] or gene information summarization[11], but there was no integration of this two related steps in microarray data analysis process.

Methods

The system is implemented in Python, and accessible via the Web. Figure 1 depicts the architecture of the system. The system's core components consists of a GO, MeSH and word pre-processor, a wrapper around CLUTO[1] (a preexisting clustering application), and a sentence ranker.

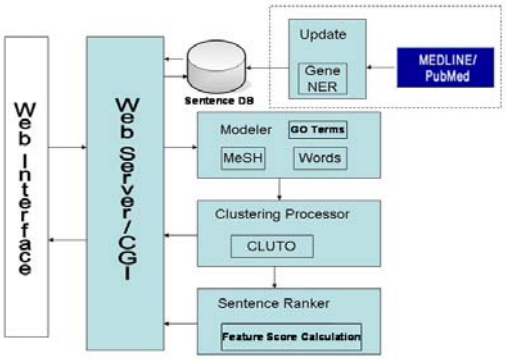

Figure 1. System architecture.

The 10-year MEDLINE corpus (from 1994 to 2003) used in TREC 2004-2005 Genomics Tracks [12] was filtered using the MeSH Heading "Mice," resulting in a mouse-focused subset. We decided to focus on mouse genes initially to allow us to restrict the domain to mouse research, an area that we had developed relationships with researchers who would participate in our study. Restricting the domain to mouse research also allowed us to achieve higher accuracy in gene name recognition. Using our gene/protein name entity recognition and normalization system (NER) [13] configured for mice, this subset was processed and gene and protein names were tagged and identified by Mouse Genome Informatics identifiers (MGI_ID). Sentences in the abstract and title were stored in a database together with other MEDLINE entries such as MeSH headings and publication date. This process resulted in the sentence collection used in this study. There are a total of 284,900 abstracts covering 11,311 mouse genes. Sentences containing at least one reference to gene/protein were further indexed by MGI_ID to facilitate retrieval.

Clustering of genes into functionally related groups. We modeled the genes by three categories of features:

1. MeSH Headings associated with the publications in which the gene is mentioned.
2. Gene Ontology (GO) terms associated with the genes as annotated by Mouse Genome Informatics (MGI) group.
3. Free text words in the sentences with at least one reference to a gene and sentences immediately before and after them, with stop word removal, and stemming.

Each gene is modeled as vector combination of the above three categories of features. Direct k-means clustering algorithm was used to find the functional clusters. Clustering was performed using these features in the following manner:

- MeSH term filtering: only terms deeper than the second layer were included because we found terms close to the root were too general, such as: 'GENE' and 'PROTEIN'.
- Similarity measure: similarity between genes is calculated as the cosine of the angle between the two gene vectors: $cos(gi, gj)=gi \bullet gj/|gi| \, |gj|$
- Number of clusters: this is a parameter CLUTO takes as input. We empirically determined this parameter at run-time. Let ε be the ratio of improvement of internal similarity of all clusters by increasing the number of cluster by 1: $\varepsilon = [I2_{(n+1)} - I2_{(n)}]/I2_{(n)}$ where I2 is the measure of sum of the internal similarity all clusters, and n is the number of clusters. Once ε reaches a low enough value, we stop. Currently it is set at 0.035, i.e., n is chosen when the improvement is less than 3.5% by increasing the number of clusters by one.

Ranking of sentences for each gene. The number of sentences for each gene identified in the literature corpus varied from one to 30,216. The users can choose the number of most recent sentences to return. Sentences are modeled as word vectors after parsing, stop word removal and stemming. Each sentence is assigned a score by linear combination of features. The features were chosen following the framework of Edmundson [5] with modifications customized to biomedical domain. Sentence score *S* is calculated as:

$$S=w_1 \, CluSim + w_2 \, NGene + w_3 \, CTword + w_4 \, TPword + w_5 \, L + w_6 \, Recency$$

[1] http://www-users.cs.umn.edu/~karypis/cluto/index.html

where *CluSim, QuFreq, NGene, CTword, TPword , L* and *Recency* are features defined below and $w_{1\text{-}6}$ are weight parameters between 0 and 1 for each feature.

- Cluster representation (*CluSim*). The top five descriptive features (a set of MeSH, GO terms and/or words) for each gene cluster from the previous step are used as this ranking measure. *CluSim* is calculated as the normalized number of feature terms the sentence has or assigned to the abstract where the sentence is extracted.
- Gene relations (*NGene*). Sentences referenced to more than one gene/protein names score higher, otherwise, 0. Emphasis on relations is also reflected in later features, such as relation words in *TPWord*.
- Cue phrases (*CTword*). This is identical to the Edmundson's Cue feature based on the assumption that the importance of a sentence is represented on the presence or the absence of certain key terms. For example, the term 'conclusion' may indicate importance.
- Domain specific keywords (*TPword*). Biologically relevant keywords were extracted from the Textpresso [14] ontology. *TPword* is calculated as count of keywords in the sentence normalized to between zero and one, with the sentence having the maximum count scoring one.
- Length (*L*). Usually the longer the sentence, the more information it contains. *L* is calculated as the faction of longest sentence.
- *Recency* is calculated as a linear scale for the sentences from one to zero, with the most recent sentence getting the score one.

There are many ways to combine the features by adjusting the weights. The weighting scheme was adjusted empirically based on feedback from users during the first stage of the evaluation process.

Evaluating clustering algorithm. Four gene sets from the result of four different microarray experiments were tested on the system by two OHSU-based mouse genomic researchers. Each person rated the gene set generated by his/her own lab. For each gene set, the participants labeled the genes they were familiar with. Each of the participants compared cluster pairs, which had at least one of the familiar labeled genes. This setup ensured each person had the expertise for the particular gene to judge the result. First, participants judged the usefulness or meaningfulness of two clusters for each gene set by comparing the clusters with random grouping of genes. Then, the effects of different clustering features (MeSH, GO, text) were evaluated by comparing clusters generated by each feature side by side. For each cluster pair, participants chose the more useful cluster of genes from the pair using a 5-point Likert scale: *1.cluster on right is absolutely better, 2. cluster on right is better, 3. they are the same, 4. cluster on left is better, and 5. cluster on left is absolutely better.* We also randomized the left/right order of the clusters.

Evaluating ranking of informative sentences. Sentences for eight genes (one from each of the cluster evaluated in the previous step) were used in this step. Sentences from the output of the system and PubMed searches (queried by the name of the gene and synonym expansion, limited to the time period 1994-2003 and filtered by MeSH term *mice.*) were pooled together and judged by the same scientists who studied the gene set. The raters assigned an R (relevant) or NR (not relevant) label to each sentence by judging if it had relevant information for understanding the specific gene studied in the microarray experiment they were analyzing. Results from two genes were used to hand-tune the ranking parameter and the other six were used to study the system. Only data from the six genes is reported here. Three ranked lists were compared by mean average precision (MAP) using the relevance judgments as a gold standard:

1. Our system output: Sentences with reference to the gene extracted from the abstracts ranked by the scoring algorithm.
2. Same sentence set as in 1 but in reversed chronological order, same as PubMed's ranking.
3. Output from PubMed search (title of abstract in reversed chronological order).

Using the relevance judgment 'gold standard', we also calculated MAP for sentences rankings using each of the single ranking features to study the separate contribution of each feature.

Results

Gene clustering. A comparison between gene clusters and random groups is shown in Figure 2. Note that a cluster received a score of '3' if it was judged as good as 'random'. Statistical analysis using one sample t-test indicated that allCluster(combining three features results) was significantly better than random at $p=0.001$. Clustering using GO terms was also significantly better at $p=0.003$, while both text and MeSH terms were not significant at $p=0.20$ and $p=0.094$.

Figure 3 shows the comparison among the three features. It seems that MeSH fared better than both GO and text terms, while GO was better than text but to a lesser extent. The differences were insignificant statistically.

Sentence ranking. The comparison for the three rankings is shown in Figure 4. Showing the sentences from the abstract did much better than titles from PubMed output. The ranking algorithm gave a 3.2% increase over simply reversed chronological order of sentences and over 50% increase above PubMed titles. Both differences were significant at *p=0.021* and *p=0.001* by pair-wise sample comparisons. Each feature's individual contribution is shown in Table 1. It appears that *TPword* was the most useful feature and *CluSim* was the least useful for sentence ranking.

Discussion

In general, the clustering algorithm gave better gene groups than random as supported by the small *p* values found when comparing all cluster results to random and GO to random. The result on MeSH and text is inconclusive. The comparison between the feature types also showed insignificant differences, even we found that MeSH and GO were better than text. We suspect it was the small sample size of this initial study that did not give us enough power. Future work should include more biologists. In addition, the system allows any combination of the features to be used for clustering, but how different combinations fare against single features was not studied here and remains for future work.

We found that judging cluster pairs was not an easy task for the scientists. Even though the cluster had at least one of the genes they choose as familiar, in order to judge the quality of the cluster, they needed to follow links in the evaluation screen for information on other genes in the cluster. It created a bigger work load for the evaluators and by the end of the session, they would make their best judgments without going through the information for not-so-familiar genes, possibly due to fatigue. The time for cluster evaluation of each gene set ranged from 20 to 35 minutes. How to best judge the quality of clusters is still in general an issue, especially in this case we define quality as how meaningful the clusters are for a specific microarray experiment, so some analytical measures, such as internal and external similarities may not correlate closely.

Providing sentences in the abstract gave much more relevant information than titles. The scoring algorithm resulted in a statistically significant higher MAP score than reversed chronological order for the same sentence set. Domain specific ontology terms improved results as indicated by the highest single feature MAP, even thought it is not significantly different from the performance by the ranking algorithm. Further research with more samples may be able to determine if TPword only can perform as well as the ranking algorithm. Interestingly, Keywords in the cluster (*cluSim*) did not seem to help much. One possible reason is that the users prefer specific information represented by the ontology terms rather than the general knowledge about the gene group functions.

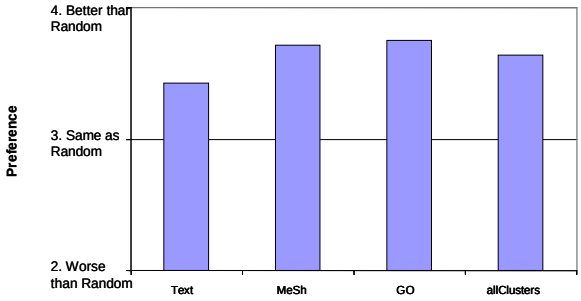

Figure 2. Comparison between clustering and random.

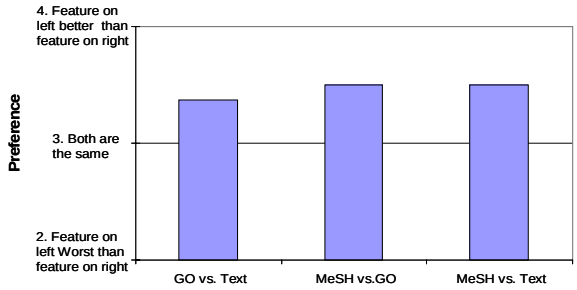

Figure 3. Cluster feature comparison.

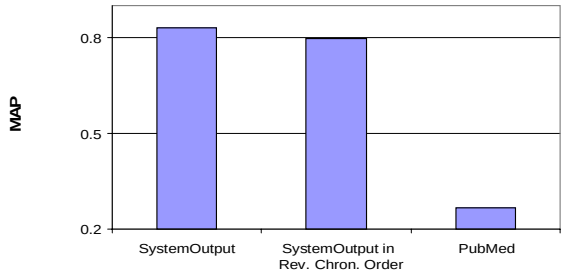

Figure 4. Mean Average Precision scores for the three ranked lists.

Features	MAP
CluSim	0.742
Ngene	0.790
CTword	0.796
TPword	0.850
Length	0.803
Recency	0.798

Table 1 MAP scores achieved by ranking with single feature only.

We used part of the evaluation data as tuning to the parameters. Since we have six parameters and use

only two examples for adjustment, we did not have enough data to fully tune the system. It is unknown how the rough tuning influenced the system performance, but the results on contribution of single features provide additional information for future tuning work.

There are several limitations to this system. First, it is built on top of a gene NER system, the accuracy of which influences the result of clustering and sentence selection. Our NER achieves state-of-the-art accuracy at 70-80% [15] but some genes got low MAP score because the sentences were in fact about other genes with identical symbols. Second, the system uses the cosine function as a measure of similarity. We will test if including semantic distance for GO and MeSH terms will improve performance[16]. Third, our system did not use more advanced NLP techniques, such as parsing and part-of-speech tagging. It will be valuable to study if adding NLP will improve the performance without the sacrifice of much speed. Finally, the document collection for the system is a static 10-year set of MEDLINE abstracts. Once the development and testing phases are completed, the database will be updated automatically once a week by download from NLM.

Conclusion

We built and evaluated a gene information summarization system for mouse genome researchers. The evaluation results indicate that our approach seems to generate meaningful gene clusters and achieve better sentence ranking than standard methods using domain specific ontology terms in addition to general sentence features.

Acknowledgements

This work was supported in part by NLM Training Grant 1T15 LM009461.

References

1. Fattore M, Arrigo P. 2004; Topical clustering of biomedical abstract by self organizing maps. In: Proceedings of 'The Fourth International Conference on Bioinformatics of Genome Regulation and Structure'; Novosibirsk, Russia, July 25-30, 2004.
2. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. 2001; GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 17(Suppl 1):S74-S82.
3. Hersh W, et al. 2006; TREC 2006 Genomics Track overview. The Fifteenth Text Retrieval Conference, TREC 2006. http://trec.nist.gov/pubs/trec15/papers/GEO06.OVERVIEW.pdf
4. Sparch Jones K. Automatic summarizing: factors and directions. In: Mani I, Maybury MT, editors. *Advances in Automatic Text Summarization.* London: MIT Press, 1999.
5. Edmundson H. 1969; New methods in automatic extracting. Journal of the ACM 16(2):264-285.
6. Luhn H. 1958; The Automatic Creation of Literature Abstracts. IBM Journal:159-165.
7. Radev D, Jing H, Stys M, Tam D. 2004; Centroid-based summarization of multiple documents. Information Processing and Management 40:919-938.
8. Teufel S, Moens M. 2002; Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. Computational Linguistics 28(4):409-445.
9. Liu Y, Ciliax BJ. 2004; Comparison of two schemes for automatic keyword extraction from MEDLINE for functional gene clustering. Proc IEEE Comput Syst Bioinform Conf:394-404.
10. Homayouni R, Heinrich K, Wei L, Berry MW. 2005; Gene clustering by Latent Semantic Indexing of MEDLINE abstracts. Bioinformatics 21(1):104-115.
11. Ling X, et al. 2005; Automatically generating gene summaries for biomedical literature. In: Proceedings of the Pacific Symposium on Bioinformatics.
12. Hersh W, et al. 2005; TREC 2005 Genomics Track overview. The Fourteenth Text Retrieval Conference, TREC 2005. http://trec.nist.gov/pubs/trec14/papers/GEO.OVERVIEW.pdf
13. Cohen A. 2005; Unsupervised gene/protein entity normalization using automatically extracted dictionaries. In: Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Proceedings of the BioLINK2005 Workshop; Detroit, MI: Association for Computational Linguistics; 2005:17-24.
14. Muller HM, Kenny EE, Sternberg PW. 2004; Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biol 2(11(e309)):1984-1998.
15. Hirschman L, Colosimo M, Morgan A, Yeh A. 2005; Overview of BioCreAtIvE task 1B: normalized gene lists. BMC Bioinformatics 6(Suppl 1):S11.
16. Wang JZ, Du Z, Payattakool R, Yu PS, Chen CF. 2007; A New Method to Measure the Semantic Similarity of GO Terms. Bioinformatics:btm087.